## REMARKS

Claims 1-18 are pending in the application. Claims 1-18 are rejected.

The title to the invention is objected to. A proposed new title is included herewith. The Examiner is invited to propose another title if this is one not acceptable.

The Abstract is objected to. A shortened and amended Abstract is included herewith.

The specification is objected to for several informalities. Corrections are included herewith. The corrections include spelling and grammar.

With respect to the paragraph 8 in the Office Action, where Fig. 1 and Fig. 14 are objected to, the specification has been amended to include the reference numbers "24" and "CHM" in pages 21 and 49. "CHM" is also supported in page 55, 2nd paragraph.

Fig. 15 is objected to for a typographical error. Enclosed is a proposed amended Fig. 15 with changes shown in red ink.

Regarding Fig. 4, enclosed is a proposed amended Fig. 4 with changes shown in red ink. To clarify the claimed invention, as shown on the enclosed mark-up of Fig. 4, the route (1) is a sequence of instructions 01-02-11-12-21-22..., the route (2) is a sequence of instructions 01-02-11-12-13...., the route (3) is a sequence of instructions 01-02-03-04-41..., and the route (4) is a sequence of instructions 01-02-03-04-05-06....

Formal substitutes will be submitted upon approval by the Examiner.

With regard to the question in paragraph 10 of the Office Action as to why branch instructions include a second (C) stage as shown in Fig. 5 for example. It is submitted that all the instructions include the stages P (pre-fetch request stage), T (cache stage), C (instruction fetching stage), D (decoding stage), E (execution stage) and W (writing stage).

No new matter is entered in the specification.

Claims 2-4, 7, 12-14, and 16 are objected to. These claims have been amended to overcome the objections.

Claims 1, 2, 6, 12, 16 are independent claims. Claims 1 and 6 have been amended to clarify applicant's claimed invention.

Claims 1, 5-9 and 11-12 are rejected under 35 U.S.C. §102(e) as being anticipated by Shiell (5,864,697). Claims 2-4 are rejected under 35 U.S.C. §103 as being unpatentable over Shiell in view of Shintani et al. (4,532,589) and further in view of Nakanishi (5,835,754). Claim 10 is rejected under 35 U.S.C. §103 as being unpatentable over Shiell in view of Hara (5,740,415). Claims 13 and 14 are rejected under 35 U.S.C. §103 as being unpatentable over Shiell in view of Shintani and further in view of the Lee et al. article "Instruction Cache Fetch policies for Speculative Execution, 1995". Claims 15-18 are rejected under 35 U.S.C. §103 as being unpatentable over Shiell in view of Lee.

Shiell teaches a pipelined microprocessor which uses combined actual and speculative branch history prediction to predict branches.

According to the invention of claim 1, the invention relates to a pre-fetch control for pre-fetching an first instruction sequence being processed and a second instruction sequence of the target address of the branching instruction in the first instruction sequence. While the first instruction sequence is being processed, the branch target address information of the branching instruction in the first instruction sequence, and the branch target address information of the branching instruction in the second instruction sequence are stored in the branch target address information buffers separately.

That is, the branch target address information buffers always stores the branch target address information for both of the first and the second instruction sequences while processing

the first instruction sequence in a separate manner. Therefore, whenever a branching instruction in the processing first instruction sequence is executed and the branching direction is determined another branch target instruction sequence can immediately be fetched and stored in the instruction buffer by using the branch target address information in the branch target address information buffer.

On the other hand, Shiell discloses a branch target buffer 56 which is similar to the BTB in Nakanishi. That is, the branch target buffer 56 stores, in a random fashion, a large number of branch target address and branch information in relation to a fetch address FA. The branch target buffer is accessed by tag bits like a cache memory. That is, in Shiell, a large number of branch target address are stored in BTB 56 in relation to the fetch address of the branch instructions. Therefore, when fetching the target address instruction based on the branch target address, the BTB 56 should be accessed based on the fetch address FA, and then the target address can be read from BTB 56. Thereafter, the target address instruction is fetched based on the read target address. This means that there need an additional access to the BTB 56.

In the present invention, the branch target address information of either the first and the second instruction sequence, which follows to a branching instruction in the first instruction sequence, are stored in the branch target information buffers separately. Therefore, such branch target address information can be obtained without accessing the branch address information buffer based on the fetch address.

Claim 2 is rejected as obvious over Shiell, Shintani and Nakanishi. Applicant respectfully disagrees in particular with paragraph 32, g, in page 14 of the Office Action. In Nakanishi, there are two BTB 11 and 21. However, these two BTB 11 and 21 do not correspond to the branch instructions in the first instruction sequence being processed and the branch

instruction in the second instruction sequence, like the present invention. Nakanishi simply discloses a method for predicting the branch direction of the fetched instruction. Nakanishi does not disclose the similar technology to the branch target address information buffer of the present invention.

According to claim 2 an information processing device which reads, buffers, decodes and executes instructions from an instruction store portion by pipeline processing, comprising: an instruction reading request portion which assigns a read address to said instruction store portion; an instruction buffering portion including a plurality of instruction buffers which buffer instruction sequences read from said instruction store portion; an instruction execution unit which decodes and executes instructions buffered by said instruction buffering portion; a branching instruction detection portion which detects a branching instruction inside the instruction sequence read from said instruction store portion; and a branch target address information buffering portion including a plurality of branch target address information buffers which, when said branching instruction detection portion has detected a branching instruction, buffer the branch target address information for generating the branch target address of said branching instruction;

wherein: a first instruction sequence (C1) being processed is stored in either one of first or second instruction buffer (e-1) and when said branching instruction detection portion detects a branching instruction inside said first instruction sequence (C1), a second instruction sequence (C2) of the branch target is stored in the other one of the first or second instruction buffers e-2) in accordance with the branch target address information of said branching instruction; the branch target address information of a next branching instruction inside said first instruction sequence (C1) is stored in either one of first or second branch target address

information buffer (b-1); and the branch target address information of the branching instruction inside said second instruction sequence (C2) is stored in the other one of the first or second branch target address information buffer (b-2).

According to claim 3, the information processing device as claimed in Claim 2 wherein, in the state in which said first instruction sequence being processed (C1) is stored in either one of said first or second instruction buffer (e-1), the second instruction sequence (C2) of the branch target of the branching instruction inside said first instruction sequence is stored in the other one of said first or second instruction buffers (e-2), the branch target address information of the next branching instruction inside said first instruction sequence (C1) is stored in said first branch target address information buffer (b-1) and the branch target address information of the branching instruction inside said second instruction sequence (C2) is stored in said second branching address information buffer (b-2); if the execution of the branching instruction inside said first instruction sequence (C1) has resulted in branching, said first instruction sequence (C1) and the branch target address information of the next branching instruction inside said first instruction sequence (C1) are invalidated; and the third instruction sequence (C4) of the branch target of the branching instruction inside said second instruction sequence (C2) is stored in one of said first or second instruction buffer (c-1), in accordance with the branch target address information which have been stored in the other one of said first or second branch target address information buffer (b-2); and the branch target address information of the next branching instruction inside said second instruction sequence (C2) is stored in one of the first or second branch target address information buffer (b-1), and the branch target address information of the branching instruction inside said third instruction sequence (C4) is stored in the other one of said first or second branch target address information buffer (b-2).

Claims 12 and 16 relate to the second embodiment shown in Figs. 14 and 15. According to the invention of these claims, when pre-fetching instruction, if there are cache miss via cache access, whether an external memory buss access for pre-fetch is preformed or not is determined in accordance with the predicted branching direction. This is because the external memory buss access needs a lengthy instruction cycles, therefore, if the pre-fetched instruction is not in the predicted branching direction, such external memory buss access is prohibited.

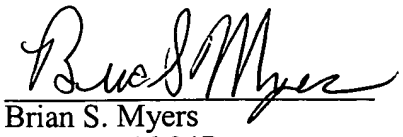
In the Office Action it's alleged that the external memory access when cache miss is occurred is inherent nature of the memory hierarchy in the paragraph 29, f, in page 11-12 in the Office Action. Applicant respectfully disagrees with this conclusion. That is, in the present invention, the instruction pre-fetch to the cache memory is performed no matter what the branching prediction is, however, if a cache miss occurred, then whether the external memory access is performed is determined in accordance with the branching prediction.

For at least the foregoing reasons it is respectfully requested the rejection of the claims be withdrawn.

In view of the remarks set forth above, this application is in condition for allowance which action is respectfully requested. However, if for any reason the Examiner should consider this application not to be in condition for allowance, the Examiner is respectfully requested to telephone the undersigned attorney at the number listed below prior to issuing a further Action.

Any fee due with this paper may be charged to Deposit Account No. 50-1290.

Respectfully submitted,



Brian S. Myers
Reg. No. 46,947

**CUSTOMER NUMBER 026304**
Katten Muchin Zavis Rosenman
Telephone: (212) 940-8703
Fax: (212) 940-8986
Docket No.: FUJH 17.759 (100794-11499)